The Word Problem for Finitary Automaton Groups

Maximilian Kotwosky[1] **Jan Philipp Wächter**[2,3]

[1]Universität Stuttgart, Institut für Formale Methoden der Informaik

[2]Politecnico di Milano, Dipartimento di Matematica

[3]funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 492814705

6 July 2023

The Word Problem of a Group

The Word Problem of a Group

- Consider a group G

The Word Problem of a Group

- Consider a group G generated by a finite set Q
  i. e. every element $g \in G$ can be written as $g = q_1^{\delta_1} \dots q_\ell^{\delta_\ell}$ with $q_i \in Q$, $\delta_i \in \{-1, 1\}$

The Word Problem of a Group

- Consider a group G generated by a finite set Q
  i. e. every element $g \in G$ can be written as $g = q_1^{\delta_1} \dots q_\ell^{\delta_\ell}$ with $q_i \in Q$, $\delta_i \in \{-1, 1\}$
- The word problem of G is the decision problem

Constant:	the group G generated by Q
Input:	a word $\boldsymbol{q} \in (Q^{\pm 1})^*$
Question:	is $\boldsymbol{q} = \mathbb{1}$ in G?

The Word Problem of a Group

- Consider a group G generated by a finite set Q
  i. e. every element $g \in G$ can be written as $g = q_1^{\delta_1} \dots q_\ell^{\delta_\ell}$ with $q_i \in Q$, $\delta_i \in \{-1, 1\}$
- The word problem of G is the decision problem

  Constant: the group G generated by Q
  Input: a word $\boldsymbol{q} \in (Q^{\pm 1})^*$
  Question: is $\boldsymbol{q} = \mathbb{1}$ in G?

- ...as a formal language: $\mathrm{WP}_Q(G) = \{\boldsymbol{q} \in (Q^{\pm 1})^* \mid \boldsymbol{q} = \mathbb{1} \text{ in } G\}$

The Word Problem of a Group

- Consider a group G generated by a finite set Q
  i. e. every element $g \in G$ can be written as $g = q_1^{\delta_1} \dots q_\ell^{\delta_\ell}$ with $q_i \in Q$, $\delta_i \in \{-1, 1\}$
- The word problem of G is the decision problem

  Constant: the group G generated by Q
  Input: a word $\boldsymbol{q} \in (Q^{\pm 1})^*$
  Question: is $\boldsymbol{q} = \mathbb{1}$ in G?

- ...as a formal language: $\mathrm{WP}_Q(G) = \{\boldsymbol{q} \in (Q^{\pm 1})^* \mid \boldsymbol{q} = \mathbb{1} \text{ in } G\}$

Fact (Anisimov 1971)

G is finite $\iff$ $\mathrm{WP}_Q(G)$ is regular

The Uniform Word Problem for Groups

But: We can also consider the group as part of the input!

The Uniform Word Problem for Groups

But: We can also consider the group as part of the input!

Definition

The uniform word problem for groups is the decision problem

Input: a group G generated by Q and
a word $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in G?

The Uniform Word Problem for Groups

But: We can also consider the group as part of the input!

Definition

The uniform word problem for groups is the decision problem

Input: a group G generated by Q and
a word $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in G?

- Problem: How can we give a group as an input to an algorithm?

The Uniform Word Problem for Groups

But: We can also consider the group as part of the input!

Definition

The uniform word problem for groups is the decision problem

Input: a group G generated by Q and
a word $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in G?

- Problem: How can we give a group as an input to an algorithm?
- Typically: using a finite presentation $G = \langle Q \mid \boldsymbol{r}_1 = \mathbb{1}, \dots, \boldsymbol{r}_k = \mathbb{1} \rangle$

The Uniform Word Problem for Groups

But: We can also consider the group as part of the input!

Definition

The uniform word problem for groups is the decision problem

Input: a group G generated by Q and
a word $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in G?

- Problem: How can we give a group as an input to an algorithm?
- Typically: using a finite presentation $G = \langle Q \mid r_1 = \mathbb{1}, \dots, r_k = \mathbb{1} \rangle$ "finitely presented"

The Uniform Word Problem for Groups

But: We can also consider the group as part of the input!

Definition

The uniform word problem for groups is the decision problem

Input: a group G generated by Q and
a word $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in G?

- Problem: How can we give a group as an input to an algorithm?
- Typically: using a finite presentation $G = \langle Q \mid r_1 = \mathbb{1}, \dots, r_k = \mathbb{1} \rangle$ "finitely presented"
- Today: We only consider finite groups!

The Uniform Word Problem for Groups

But: We can also consider the group as part of the input!

Definition

The uniform word problem for groups is the decision problem

Input: a group G generated by Q and
a word $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in G?

- Problem: How can we give a group as an input to an algorithm?
- Typically: using a finite presentation $G = \langle Q \mid r_1 = \mathbb{1}, \dots, r_k = \mathbb{1} \rangle$ "finitely presented"
- Today: We only consider finite groups! Possible: $Q = G$

The Uniform Word Problem for Groups

But: We can also consider the group as part of the input!

Definition

The uniform word problem for groups is the decision problem

Input: a group G generated by Q and
a word $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in G?

- Problem: How can we give a group as an input to an algorithm?
- Typically: using a finite presentation $G = \langle Q \mid r_1 = \mathbb{1}, \ldots, r_k = \mathbb{1} \rangle$ "finitely presented"
- Today: We only consider finite groups! Possible: $Q = G$
- Possible descriptions: Cayley tables, Cayley graphs, matrices, permutations, ...

Some Known Results: Upper Bounds

Some Known Results: Upper Bounds

Fact

The word problem for groups given as Cayley tables

Input: *a Cayley table* $G \times G \to G, (g, h) \mapsto gh$ *of a finite group* G *and group elements* $g_1, \dots, g_n \in G$

Question: *is* $g_1 \cdot \ldots \cdot g_n = \mathbb{1}$*?*

is in LogSpace.

Some Known Results: Upper Bounds

Fact

The word problem for groups given as Cayley tables

Input: a Cayley table $G \times G \to G, (g, h) \mapsto gh$ of a finite group G and group elements $g_1, \dots, g_n \in G$

Question: is $g_1 \cdot \ldots \cdot g_n = \mathbb{1}$?

is in $\textsc{LogSpace}$.

Theorem (Lipton, Zalcstein 1977/Simon 1979)

The word problem of a finitely generated linear group

Constant: $G \leq \mathrm{GL}(d, \mathbb{F})$

Input: matrices $M_1, \dots, M_n \in G$

Question: is $M_1 \cdot \ldots \cdot M_n$ the identity matrix?

is in $\textsc{LogSpace}$.

Some Known Results: Lower Bounds

Some Known Results: Lower Bounds

Theorem (Cook, McKenzie 1987)

The problem

Input: *permutations* $\pi_1, \ldots, \pi_\ell$ *in cycle notation*

Output: *the product* $\pi_1 \ldots \pi_\ell$ *in cycle notation*

is complete for functional LogSpace.

Some Known Results: Lower Bounds

Theorem (Cook, McKenzie 1987)

The problem

Input: *permutations* $\pi_1, \dots, \pi_\ell$ *in cycle notation*

Output: *the product* $\pi_1 \dots \pi_\ell$ *in cycle notation*

is complete for functional $\textsc{LogSpace}$*.*

Theorem (Barrington 1986)

The word problem $\mathrm{WP}(A_5)$ *of the group of even permutations over* $\{a_1, \dots, a_5\}$ *is* NC^1*-complete.*

Some Known Results: Lower Bounds

Theorem (Cook, McKenzie 1987)

The problem

Input: *permutations* $\pi_1, \dots, \pi_\ell$ *in cycle notation*
Output: *the product* $\pi_1 \dots \pi_\ell$ *in cycle notation*

is complete for functional $\textsc{LogSpace}$.

Theorem (Barrington 1986)

The word problem $\mathrm{WP}(A_5)$ *of the group of even permutations over* $\{a_1, \dots, a_5\}$ *is* NC^1*-complete.*

Boolean circuits, bounded fan-in, $O(\log n)$ depth;
$\mathrm{NC}^1 \subseteq \textsc{LogSpace}$

Some Known Results: Lower Bounds

Theorem (Cook, McKenzie 1987)

The problem

Input: *permutations* $\pi_1, \ldots, \pi_\ell$ *in cycle notation*

Output: *the product* $\pi_1 \ldots \pi_\ell$ *in cycle notation*

is complete for functional LogSpace.

Theorem (Barrington 1986)

The word problem $\mathrm{WP}(A_5)$ *of the group of even permutations over* $\{a_1, \ldots, a_5\}$ *is* NC^1*-complete.* Boolean circuits, bounded fan-in, $\mathcal{O}(\log n)$ depth; $\mathrm{NC}^1 \subseteq$ LogSpace

In fact: this holds for any non-solvable finite group!

Some Known Results: Lower Bounds

Theorem (Cook, McKenzie 1987)

The problem

Input: *permutations* $\pi_1, \dots, \pi_\ell$ *in cycle notation*
Output: *the product* $\pi_1 \dots \pi_\ell$ *in cycle notation*

is complete for functional $\textsc{LogSpace}$*.*

Theorem (Barrington 1986)

The word problem $\mathrm{WP}(A_5)$ *of the group of even permutations over* $\{a_1, \dots, a_5\}$ *is* NC^1*-complete.* Boolean circuits, bounded fan-in, $\mathcal{O}(\log n)$ depth; $\mathrm{NC}^1 \subseteq \textsc{LogSpace}$

In fact: this holds for any non-solvable finite group!
This yields: The uniform word problem for any group presentation (allowing A_5) is NC^1-hard!

Presenting Groups Using Automata

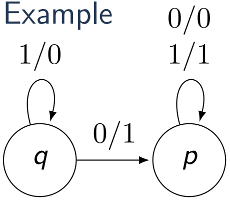
Automata

- In this setting, a $\mathscr{G}$-automaton is a
  - finite-state,
  - letter-to-letter

  transducer
  - without final or initial states

  which is
  - complete,
  - deterministic and
  - invertible.

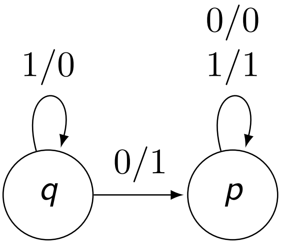
Example



State Actions

- Idea: every state q induces a bijection $\Sigma^* \to \Sigma^*$ mapping input to output words

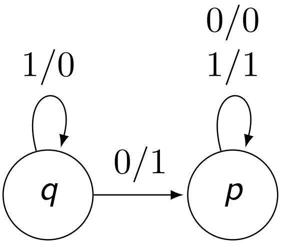
Example



State Actions

- Idea: every state q induces a bijection $\Sigma^* \to \Sigma^*$ mapping input to output words

Example

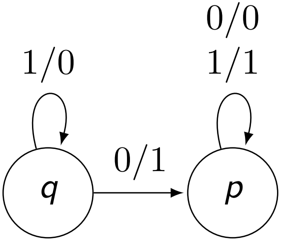


- p induces the identity function

State Actions

- Idea: every state q induces a bijection $\Sigma^* \to \Sigma^*$ mapping input to output words

Example

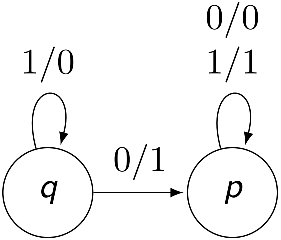


- p induces the identity function
- $$q \xrightarrow[1]{0} p \xrightarrow[0]{0} p \xrightarrow[0]{0} p \qquad q \circ 000 = 100$$

State Actions

- Idea: every state q induces a bijection $\Sigma^* \to \Sigma^*$ mapping input to output words

Example

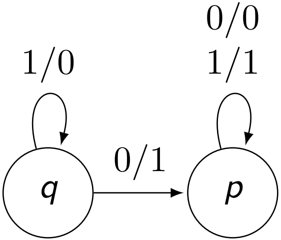


- p induces the identity function
- $$\begin{array}{ccccccc} & 0 & & 0 & & 0 & \\ q & \to & p & \to & p & \to & p \\ & 1 & & 0 & & 0 & \\ q & \to & q & \to & p & \to & p \\ & 0 & & 1 & & 0 & \end{array}$$
  $$q \circ 000 = 100$$
  $$q \circ 100 = 010$$

State Actions

- Idea: every state q induces a bijection $\Sigma^* \to \Sigma^*$ mapping input to output words

Example

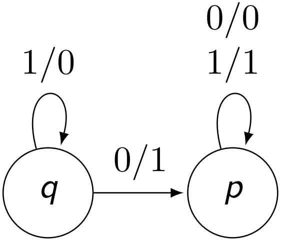


- p induces the identity function
- $$\begin{array}{ccccccc} & 0 & & 0 & & 0 & \\ q & \xrightarrow{} & p & \xrightarrow{} & p & \xrightarrow{} & p \\ & 1 & & 0 & & 0 & \\ q & \xrightarrow{} & q & \xrightarrow{} & p & \xrightarrow{} & p \\ & 0 & & 1 & & 0 & \end{array}$$
  $$q \circ 000 = 100$$
  $$q \circ 100 = 010$$
  $$q \circ 010 = 110$$

State Actions

- Idea: every state q induces a bijection $\Sigma^* \to \Sigma^*$ mapping input to output words

Example



- p induces the identity function
- $$\begin{array}{ccccccc} & 0 & & 0 & & 0 & \\ q & \longrightarrow & p & \longrightarrow & p & \longrightarrow & p \\ & 1 & & 0 & & 0 & \\ q & \longrightarrow & q & \longrightarrow & p & \longrightarrow & p \\ & 0 & & 1 & & 0 & \end{array}$$
  $$q \circ 000 = 100$$
  $$q \circ 100 = 010$$
  $$q \circ 010 = 110$$

  $\rightsquigarrow$ q increments (reverse) binary representation (least significant bit first)

Automaton Groups

- A $\mathscr{G}$-automaton $\mathcal{T}$ with state set Q generates a group $\mathscr{G}(\mathcal{T})$:

Automaton Groups

- A $\mathscr{G}$-automaton $\mathcal{T}$ with state set Q generates a group $\mathscr{G}(\mathcal{T})$:
  it is the closure under composition of the bijections induced by the states and their inverses.

Automaton Groups

Such a group is an automaton group.

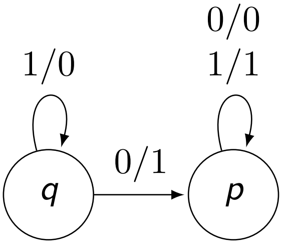
- A $\mathscr{G}$-automaton $\mathcal{T}$ with state set Q generates a group $\mathscr{G}(\mathcal{T})$:
  it is the closure under composition of the bijections induced by the states and their inverses.

Automaton Groups

Such a group is an automaton group.

- A $\mathscr{G}$-automaton $\mathcal{T}$ with state set Q generates a group $\mathscr{G}(\mathcal{T})$:
  it is the closure under composition of the bijections induced by the states and their inverses.

Example



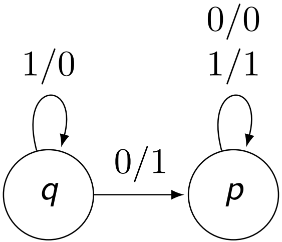
- p: identity
- q: increment

Automaton Groups

Such a group is an automaton group.

- A $\mathscr{G}$-automaton $\mathcal{T}$ with state set Q generates a group $\mathscr{G}(\mathcal{T})$:
  it is the closure under composition of the bijections induced by the states and their inverses.

Example



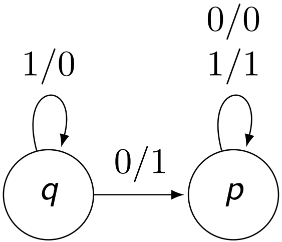
- p: identity
- q: increment
- $qp = pq = q$ in $\mathscr{G}(\mathcal{T})$

Automaton Groups

Such a group is an automaton group.

- A $\mathscr{G}$-automaton $\mathcal{T}$ with state set Q generates a group $\mathscr{G}(\mathcal{T})$:
  it is the closure under composition of the bijections induced by the states and their inverses.

Example



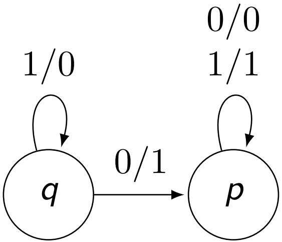
- p: identity
- q: increment
- $qp = pq = q$ in $\mathscr{G}(\mathcal{T})$
- $qq \circ 000 = q \circ 100 = 010$
- q^n: "add n"

Automaton Groups

Such a group is an automaton group.

- A $\mathscr{G}$-automaton $\mathcal{T}$ with state set Q generates a group $\mathscr{G}(\mathcal{T})$:
  it is the closure under composition of the bijections induced by the states and their inverses.

Example



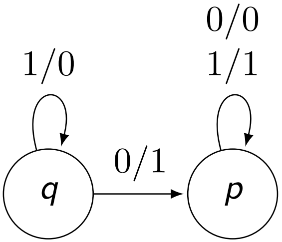
- p: identity
- q: increment, q^{-1}: decrement
- $qp = pq = q$ in $\mathscr{G}(\mathcal{T})$
- $qq \circ 000 = q \circ 100 = 010$
- q^n: "add n", q^{-n}: "subtract n"

Automaton Groups

Such a group is an automaton group.

- A $\mathscr{G}$-automaton $\mathcal{T}$ with state set Q generates a group $\mathscr{G}(\mathcal{T})$:
  it is the closure under composition of the bijections induced by the states and their inverses.

Example



- p: identity
- q: increment, q^{-1}: decrement
- $qp = pq = q$ in $\mathscr{G}(\mathcal{T})$
- $qq \circ 000 = q \circ 100 = 010$
- q^n: "add n", q^{-n}: "subtract n"

$$\mathscr{G}(\mathcal{T}) = F(q) \simeq \mathbb{Z}$$

Finitary Automaton Groups

Finitary Automaton Groups

A finitary automaton has no cycles except for self-loops at the identity state

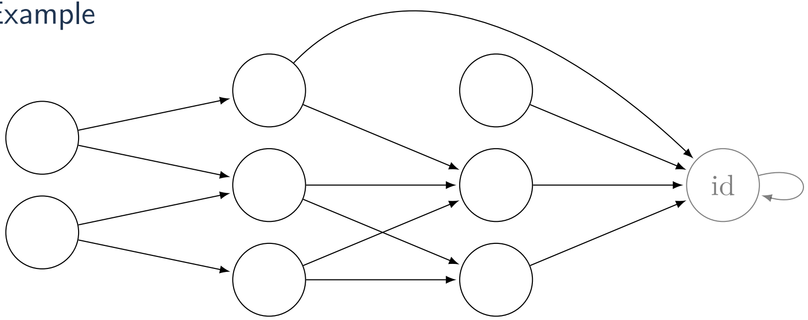
Finitary Automaton Groups

A finitary automaton has no cycles except for self-loops at the identity state
$\rightsquigarrow$ it is a labeled directed acyclic graph

Finitary Automaton Groups

A finitary automaton has no cycles except for self-loops at the identity state
$\rightsquigarrow$ it is a labeled directed acyclic graph

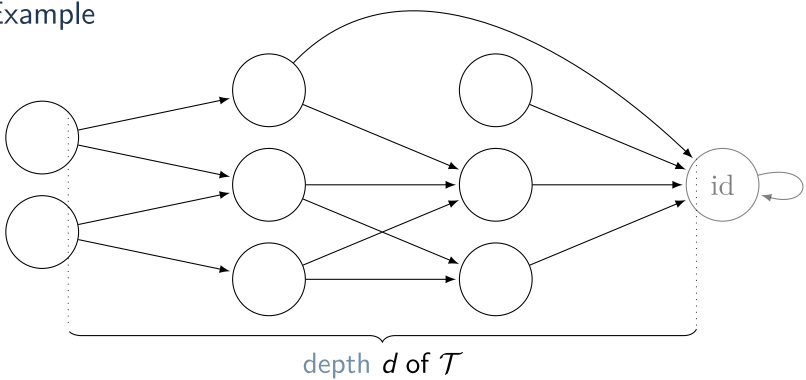
Example



Finitary Automaton Groups

A finitary automaton has no cycles except for self-loops at the identity state
⇝ it is a labeled directed acyclic graph

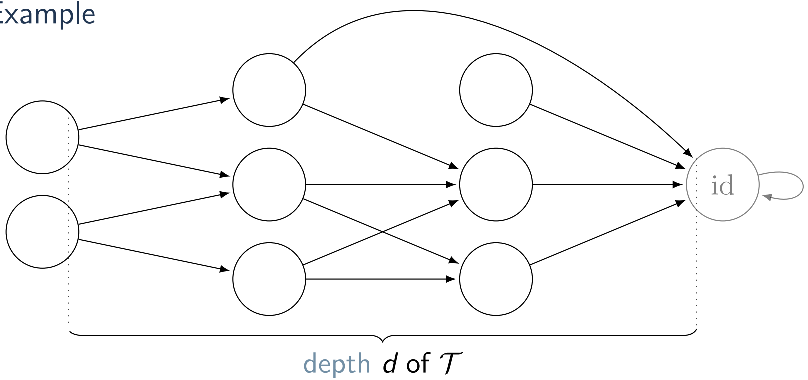
Example



Finitary Automaton Groups

A finitary automaton has no cycles except for self-loops at the identity state
$\rightsquigarrow$ it is a labeled directed acyclic graph

Example

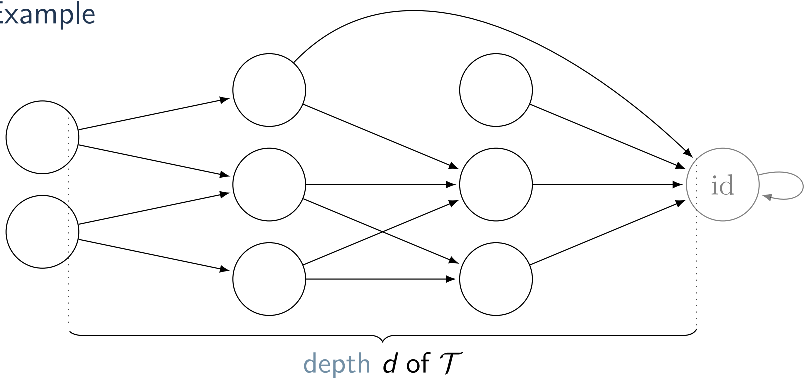


Effectively all functions are $\Sigma^d \to \Sigma^d$

Finitary Automaton Groups as Finite Groups

A finitary automaton has no cycles except for self-loops at the identity state
⇝ it is a labeled directed acyclic graph

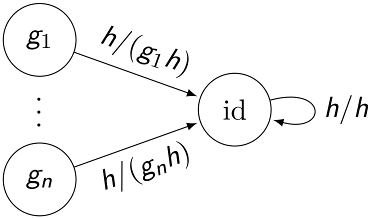
Example



Effectively all functions are $\Sigma^d \to \Sigma^d$
⇝ all finitary automaton groups are finite

Finite Groups as Finitary Automaton Groups

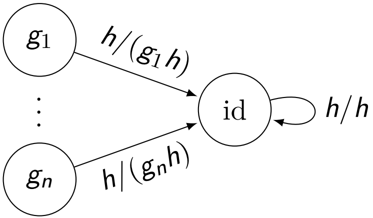
An arbitrary finite group $G = \{\mathrm{id}, g_1, \dots, g_n\}$ is generated by the finitary $\mathscr{G}$-automaton



$h \in G$

Finite Groups as Finitary Automaton Groups

An arbitrary finite group $G = \{\mathrm{id}, g_1, \ldots, g_n\}$ is generated by the finitary $\mathscr{G}$-automaton



$h \in G$

Fact

G *is finite* $\iff$ G *is a finitary automaton group*

Why Automata?

Why Automata?

Because: the presentation using automata is powerful

Why Automata?

Because: the presentation using automata is powerful

- General case: Many groups with interesting properties are automaton groups

Why Automata?

Because: the presentation using automata is powerful

- General case: Many groups with interesting properties are automaton groups
  For Example: Grigorchuk's group

Why Automata?

Because: the presentation using automata is powerful

- General case: Many groups with interesting properties are automaton groups
  For Example: Grigorchuk's group, which is not finitely presented.

Why Automata?

Because: the presentation using automata is powerful

- General case: Many groups with interesting properties are automaton groups
  For Example: Grigorchuk's group, which is not finitely presented.
  $\rightsquigarrow$ finite automata can encode groups without traditional finite presentations

Why Automata?

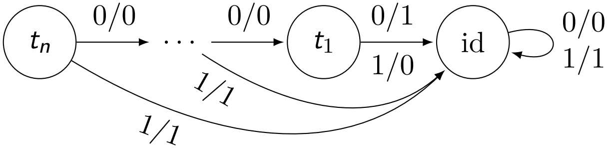
Because: the presentation using automata is powerful

- General case: Many groups with interesting properties are automaton groups
  For Example: Grigorchuk's group, which is not finitely presented.
  $\rightsquigarrow$ finite automata can encode groups without traditional finite presentations
- For finite groups: We can achieve a doubly exponential compression

Why Automata?

Because: the presentation using automata is powerful

- General case: Many groups with interesting properties are automaton groups
  For Example: Grigorchuk's group, which is not finitely presented.
  $\rightsquigarrow$ finite automata can encode groups without traditional finite presentations
- For finite groups: We can achieve a doubly exponential compression
  For Example:

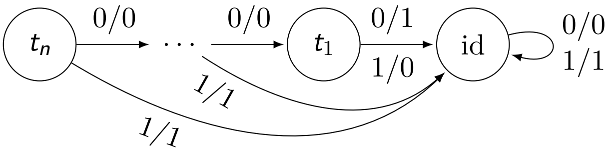


$|\mathcal{T}| = n + 1$

Why Automata?

Because: the presentation using automata is powerful

- General case: Many groups with interesting properties are automaton groups
  For Example: Grigorchuk's group, which is not finitely presented.
  $\rightsquigarrow$ finite automata can encode groups without traditional finite presentations
- For finite groups: We can achieve a doubly exponential compression
  For Example:



Automorphism group of the regular binary tree of depth n

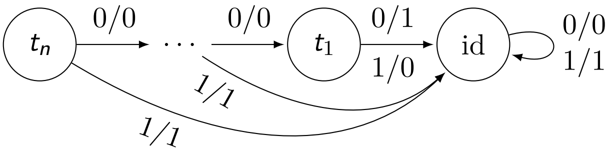
$|\mathcal{T}| = n + 1$

$\mathscr{G}(\mathcal{T}) = \operatorname{Aut} B_n$

Why Automata?

Because: the presentation using automata is powerful

- General case: Many groups with interesting properties are automaton groups
  For Example: Grigorchuk's group, which is not finitely presented.
  $\rightsquigarrow$ finite automata can encode groups without traditional finite presentations
- For finite groups: We can achieve a doubly exponential compression
  For Example:



Automorphism group of the regular binary tree of depth n

$$|\mathcal{T}| = n + 1$$
$$\mathscr{G}(\mathcal{T}) = \operatorname{Aut} B_n$$
$$\implies |\mathscr{G}(\mathcal{T})| = 2^{2^n - 1}$$

The Uniform Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform word problem for finitary automaton groups

Input: a *finitary* $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
$\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} \circ u = u$ for all $u \in \Sigma^*$ (i. e. $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$)?

is $\textsc{coNP}$*-complete.*

The Uniform Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform word problem for finitary automaton groups

Input: a *finitary* $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
$\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} \circ u = u$ for all $u \in \Sigma^*$ (i. e. $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$)?

is coNP*-complete.*

…it is PSpace-complete for general automaton groups
W., Weiß (2020)

The Uniform Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform word problem for finitary automaton groups

Input: a *finitary* $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
$\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} \circ u = u$ for all $u \in \Sigma^*$ (i. e. $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$)?

is coNP*-complete.*

…it is PSpace-complete for general automaton groups
W., Weiß (2020)

Proof (complement is in NP).

The Uniform Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform word problem for finitary automaton groups

Input: a *finitary* $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
$\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} \circ u = u$ for all $u \in \Sigma^*$ (i. e. $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$)?

is coNP*-complete.*

...it is PSpace-complete for general automaton groups
W., Weiß (2020)

Proof (complement is in NP).

- For the depth $d < |Q|$, we have:

$$\boldsymbol{q} \xrightarrow[v]{u} \mathrm{id}^{|\boldsymbol{q}|} \quad \text{for all } u \in \Sigma^{\geq d}.$$

The Uniform Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform word problem for finitary automaton groups

Input: a *finitary* $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
$\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} \circ u = u$ for all $u \in \Sigma^*$ (i. e. $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$)?

is coNP*-complete.*

...it is PSpace-complete for general automaton groups
W., Weiß (2020)

Proof (complement is in NP).

- For the depth $d < |Q|$, we have:

$$\boldsymbol{q} \xrightarrow[v]{u} \mathrm{id}^{|\boldsymbol{q}|} \quad \text{for all } u \in \Sigma^{\geq d}.$$

- $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T}) \implies \exists u \in \Sigma^d : \boldsymbol{q} \circ u \neq u$

The Uniform Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform word problem for finitary automaton groups

Input: a *finitary* $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
$\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} \circ u = u$ for all $u \in \Sigma^*$ (i. e. $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$)?

is coNP*-complete.*

…it is PSpace-complete for general automaton groups
W., Weiß (2020)

Proof (complement is in NP).

- For the depth $d < |Q|$, we have:

$$\boldsymbol{q} \xrightarrow[v]{u} \mathrm{id}^{|\boldsymbol{q}|} \quad \text{for all } u \in \Sigma^{\geq d}.$$

- $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T}) \implies \exists u \in \Sigma^d : \boldsymbol{q} \circ u \neq u$
- Algorithm: "guess & check"

The Uniform Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform word problem for finitary automaton groups

Input: a *finitary* $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
$\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} \circ u = u$ for all $u \in \Sigma^*$ (i. e. $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$)?

is coNP*-complete.*

…it is PSpace-complete for general automaton groups
W., Weiß (2020)

Proof (complement is in NP).

- For the depth $d < |Q|$, we have:

$$\boldsymbol{q} \xrightarrow[v]{u} \mathrm{id}^{|\boldsymbol{q}|} \quad \text{for all } u \in \Sigma^{\geq d}.$$

- $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T}) \implies \exists u \in \Sigma^d : \boldsymbol{q} \circ u \neq u$
- Algorithm: "guess & check"
  - Guess witness u with $|u| < |Q|$ (in time $|Q|$).

The Uniform Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform word problem for finitary automaton groups

Input: a *finitary* $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
$\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} \circ u = u$ for all $u \in \Sigma^*$ (i. e. $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$)?

is coNP*-complete.*

…it is PSpace-complete for general automaton groups
W., Weiß (2020)

Proof (complement is in NP).

- For the depth $d < |Q|$, we have:

$$\boldsymbol{q} \xrightarrow[v]{u} \mathrm{id}^{|\boldsymbol{q}|} \quad \text{for all } u \in \Sigma^{\geq d}.$$

- $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T}) \implies \exists u \in \Sigma^d : \boldsymbol{q} \circ u \neq u$
- Algorithm: "guess & check"
  - Guess witness u with $|u| < |Q|$ (in time $|Q|$).
  - Check $\boldsymbol{q} \circ u \neq u$ (in time $\approx |Q| \cdot |\boldsymbol{q}|$).

Barrington's Idea (1986)

A_5: Group of even permutations over $\{a_1, \dots, a_5\}$

Barrington's Idea (1986)

A_5: Group of even permutations over $\{a_1, \ldots, a_5\}$

Fact $\sigma^\alpha = \alpha^{-1}\sigma\alpha$

There are $\sigma, \alpha, \beta \in A_5$ *with* $\sigma \neq \mathrm{id}$ *and* $\sigma = [\sigma^\beta, \sigma^\alpha]$.

$[h, g] = h^{-1}g^{-1}hg$

Barrington's Idea (1986)

A_5: Group of even permutations over $\{a_1, \ldots, a_5\}$

Fact

$\sigma^\alpha = \alpha^{-1}\sigma\alpha$

There are $\sigma, \alpha, \beta \in A_5$ with $\sigma \neq \mathrm{id}$ and $\sigma = [\sigma^\beta, \sigma^\alpha]$.

$[h, g] = h^{-1}g^{-1}hg$

Definition (Balanced Commutator)

$$B[\boldsymbol{q}_1] = \boldsymbol{q}_1$$

$$B[\boldsymbol{q}_t, \ldots, \boldsymbol{q}_1] = \left[B[\boldsymbol{q}_t, \ldots, \boldsymbol{q}_{\lfloor \frac{t}{2} \rfloor + 1}]^\beta,\ B[\boldsymbol{q}_{\lfloor \frac{t}{2} \rfloor}, \ldots, \boldsymbol{q}_1]^\alpha\right]$$

Barrington's Idea (1986)

A_5: Group of even permutations over $\{a_1, \ldots, a_5\}$

It's a logical conjunction!

Fact $\sigma^\alpha = \alpha^{-1}\sigma\alpha$

There are $\sigma, \alpha, \beta \in A_5$ with $\sigma \neq \mathrm{id}$ and $\sigma = [\sigma^\beta, \sigma^\alpha]$.

$[h, g] = h^{-1}g^{-1}hg$

Proposition

$g_1, \ldots, g_t \in \{\sigma, \mathrm{id}\}$

$$B[g_t, \ldots, g_1] = \begin{cases} \sigma & \forall i : g_i = \sigma \\ \mathrm{id} & \text{otherwise} \end{cases}$$

Definition (Balanced Commutator)

$$B[\boldsymbol{q}_1] = \boldsymbol{q}_1$$

$$B[\boldsymbol{q}_t, \ldots, \boldsymbol{q}_1] = \left[B[\boldsymbol{q}_t, \ldots, \boldsymbol{q}_{\lfloor \frac{t}{2} \rfloor + 1}]^\beta, \; B[\boldsymbol{q}_{\lfloor \frac{t}{2} \rfloor}, \ldots, \boldsymbol{q}_1]^\alpha \right]$$

Barrington's Idea (1986)

A_5: Group of even permutations over $\{a_1, \ldots, a_5\}$

It's a logical conjunction!

Fact

$\sigma^\alpha = \alpha^{-1}\sigma\alpha$

There are $\sigma, \alpha, \beta \in A_5$ *with* $\sigma \neq \mathrm{id}$ *and* $\sigma = [\sigma^\beta, \sigma^\alpha]$.

$[h, g] = h^{-1}g^{-1}hg$

Definition (Balanced Commutator)

$$B[\boldsymbol{q}_1] = \boldsymbol{q}_1$$
$$B[\boldsymbol{q}_t, \ldots, \boldsymbol{q}_1] = \left[B[\boldsymbol{q}_t, \ldots, \boldsymbol{q}_{\lfloor \frac{t}{2} \rfloor + 1}]^\beta,\ B[\boldsymbol{q}_{\lfloor \frac{t}{2} \rfloor}, \ldots, \boldsymbol{q}_1]^\alpha\right]$$

Proposition

$g_1, \ldots, g_t \in \{\sigma, \mathrm{id}\}$

$$B[g_t, \ldots, g_1] = \begin{cases} \sigma & \forall i : g_i = \sigma \\ \mathrm{id} & \text{otherwise} \end{cases}$$

Proposition

$B[\boldsymbol{q}_t, \ldots, \boldsymbol{q}_1]$ can be computed in LogSpace.

Proof (complement is NP-hard)

Proof (complement is NP-hard)

- We reduce $3\mathrm{SAT}$

  Input: boolean formula $\varphi = \bigwedge_{k=1}^{K} C_k$ with
  $C_k = (\neg)X_{n_{k,3}} \vee (\neg)X_{n_{k,2}} \vee (\neg)X_{n_{k,1}}$ over variables $\mathbb{X} = \{X_1, \dots, X_N\}$

  Question: $\exists \mathcal{A} : \mathbb{X} \to \mathbb{B} : \mathcal{A} \models \varphi$?

  to the complement of the word problem.

Proof (complement is NP-hard)

- We reduce $3\mathrm{SAT}$

  Input: boolean formula $\varphi = \bigwedge_{k=1}^{K} C_k$ with
  $C_k = (\neg)X_{n_{k,3}} \vee (\neg)X_{n_{k,2}} \vee (\neg)X_{n_{k,1}}$ over variables $\mathbb{X} = \{X_1, \dots, X_N\}$

  Question: $\exists \mathcal{A} : \mathbb{X} \to \mathbb{B} : \mathcal{A} \models \varphi$?

  to the complement of the word problem.
- We need a map $\varphi \mapsto (\mathcal{T}, \boldsymbol{q})$ in logarithmic space s. t. φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Proof (complement is NP-hard)

- We reduce $3\mathrm{SAT}$

  Input: boolean formula $\varphi = \bigwedge_{k=1}^{K} C_k$ with
  $C_k = (\neg)X_{n_{k,3}} \vee (\neg)X_{n_{k,2}} \vee (\neg)X_{n_{k,1}}$ over variables $\mathbb{X} = \{X_1, \ldots, X_N\}$

  Question: $\exists \mathcal{A} : \mathbb{X} \to \mathbb{B} : \mathcal{A} \models \varphi$?

  to the complement of the word problem.
- We need a map $\varphi \mapsto (\mathcal{T}, \boldsymbol{q})$ in logarithmic space s. t. φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$
- alphabet: $\Sigma = \{a_1, \ldots, a_5\}$

Proof (complement is NP-hard)

- We reduce $3\mathrm{SAT}$

  Input: boolean formula $\varphi = \bigwedge_{k=1}^{K} C_k$ with $C_k = (\neg)X_{n_{k,3}} \vee (\neg)X_{n_{k,2}} \vee (\neg)X_{n_{k,1}}$ over variables $\mathbb{X} = \{X_1, \dots, X_N\}$

  Question: $\exists \mathcal{A} : \mathbb{X} \to \mathbb{B} : \mathcal{A} \models \varphi$?

  to the complement of the word problem.
- We need a map $\varphi \mapsto (\mathcal{T}, \boldsymbol{q})$ in logarithmic space s. t. φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$
- alphabet: $\Sigma = \{a_1, \dots, a_5\} \ni \bot, \top$

Proof (complement is NP-hard)

- We reduce $3\mathrm{SAT}$

  Input: boolean formula $\varphi = \bigwedge_{k=1}^{K} C_k$ with
  $C_k = (\neg)X_{n_{k,3}} \vee (\neg)X_{n_{k,2}} \vee (\neg)X_{n_{k,1}}$ over variables $\mathbb{X} = \{X_1, \dots, X_N\}$

  Question: $\exists \mathcal{A} : \mathbb{X} \to \mathbb{B} : \mathcal{A} \models \varphi$?

  to the complement of the word problem.
- We need a map $\varphi \mapsto (\mathcal{T}, \boldsymbol{q})$ in logarithmic space s. t. φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$
- alphabet: $\Sigma = \{a_1, \dots, a_5\} \ni \bot, \top$ $\langle \mathcal{A} \rangle \in \{\bot, \top\}^N$: encoding of $\mathcal{A}$

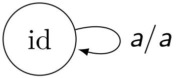
Proof (complement is NP-hard)

- We reduce $3\mathrm{SAT}$

  Input: boolean formula $\varphi = \bigwedge_{k=1}^{K} C_k$ with
  $C_k = (\neg)X_{n_{k,3}} \vee (\neg)X_{n_{k,2}} \vee (\neg)X_{n_{k,1}}$ over variables $\mathbb{X} = \{X_1, \ldots, X_N\}$

  Question: $\exists \mathcal{A} : \mathbb{X} \to \mathbb{B} : \mathcal{A} \models \varphi$?

  to the complement of the word problem.
- We need a map $\varphi \mapsto (\mathcal{T}, \boldsymbol{q})$ in logarithmic space s. t. φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$
- alphabet: $\Sigma = \{a_1, \ldots, a_5\} \ni \bot, \top$ $\langle \mathcal{A} \rangle \in \{\bot, \top\}^N$: encoding of $\mathcal{A}$
- technical states:



for all $a \in \Sigma$

Proof (complement is NP-hard)

- We reduce $3\mathrm{SAT}$

  Input: boolean formula $\varphi = \bigwedge_{k=1}^{K} C_k$ with
  $C_k = (\neg)X_{n_{k,3}} \vee (\neg)X_{n_{k,2}} \vee (\neg)X_{n_{k,1}}$ over variables $\mathbb{X} = \{X_1, \ldots, X_N\}$

  Question: $\exists \mathcal{A} : \mathbb{X} \to \mathbb{B} : \mathcal{A} \models \varphi$?

  to the complement of the word problem.
- We need a map $\varphi \mapsto (\mathcal{T}, \boldsymbol{q})$ in logarithmic space s. t. φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$
- alphabet: $\Sigma = \{a_1, \ldots, a_5\} \ni \bot, \top$ $\langle \mathcal{A} \rangle \in \{\bot, \top\}^N$: encoding of $\mathcal{A}$
- technical states:

$$\begin{tikzcd} \alpha_N \arrow[r, "a/a"] & \alpha_{N-1} \arrow[r, "a/a"] & \cdots \arrow[r, "a/a"] & \alpha_0 \arrow[r, "a/\alpha(a)"] & \mathrm{id} \arrow[loop right, "a/a"] \end{tikzcd} \quad \text{for all } a \in \Sigma$$

Proof (complement is NP-hard)

- We reduce 3SAT

  Input: boolean formula $\varphi = \bigwedge_{k=1}^{K} C_k$ with
  $C_k = (\neg)X_{n_{k,3}} \vee (\neg)X_{n_{k,2}} \vee (\neg)X_{n_{k,1}}$ over variables $\mathbb{X} = \{X_1, \dots, X_N\}$

  Question: $\exists \mathcal{A} : \mathbb{X} \to \mathbb{B} : \mathcal{A} \models \varphi$?

  to the complement of the word problem.
- We need a map $\varphi \mapsto (\mathcal{T}, \boldsymbol{q})$ in logarithmic space s. t. φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$
- alphabet: $\Sigma = \{a_1, \dots, a_5\} \ni \bot, \top$ $\qquad \langle \mathcal{A} \rangle \in \{\bot, \top\}^N$: encoding of $\mathcal{A}$
- technical states:

$$\alpha_N \xrightarrow{a/a} \alpha_{N-1} \xrightarrow{a/a} \cdots \xrightarrow{a/a} \alpha_0 \xrightarrow{a/\alpha(a)} \mathrm{id} \circlearrowleft a/a \quad \text{for all } a \in \Sigma$$

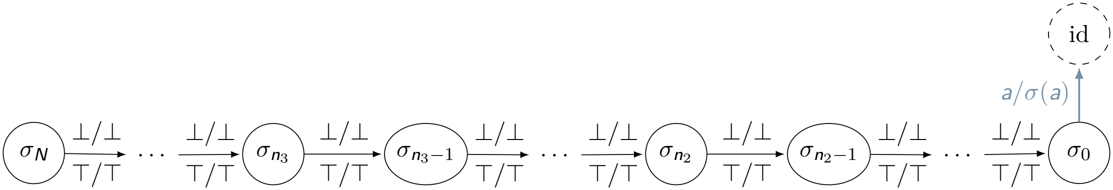
- same for β

Proof (continued)

- Important part:

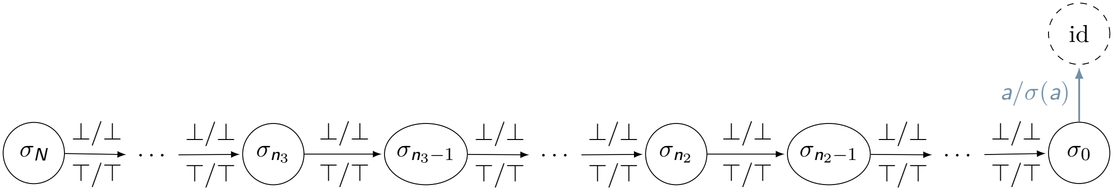
Proof (continued)

- Important part:



Proof (continued)

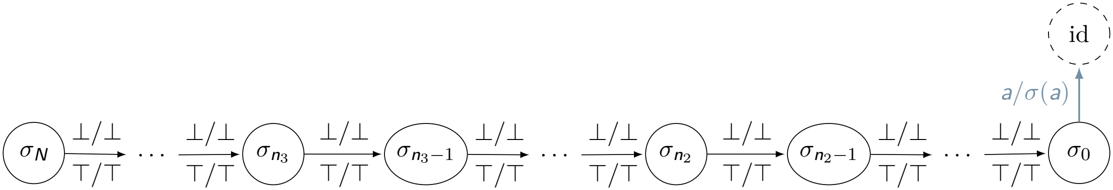
- Important part:



missing transition go to id with b/b

Proof (continued)

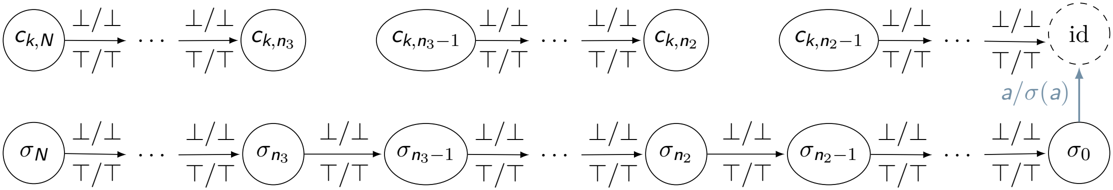
- Important part: **Example:** $C_k = X_{n_3} \vee \neg X_{n_2} \vee X_{n_1}$ (w. l. o. g.: $n_3 < n_2 < n_1$)



missing transition go to id with b/b

Proof (continued)

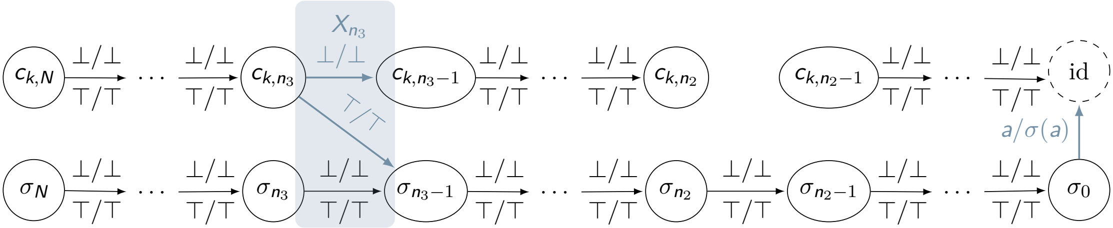
- Important part: **Example:** $C_k = X_{n_3} \vee \neg X_{n_2} \vee X_{n_1}$ (w. l. o. g.: $n_3 < n_2 < n_1$)



missing transition go to id with b/b

Proof (continued)

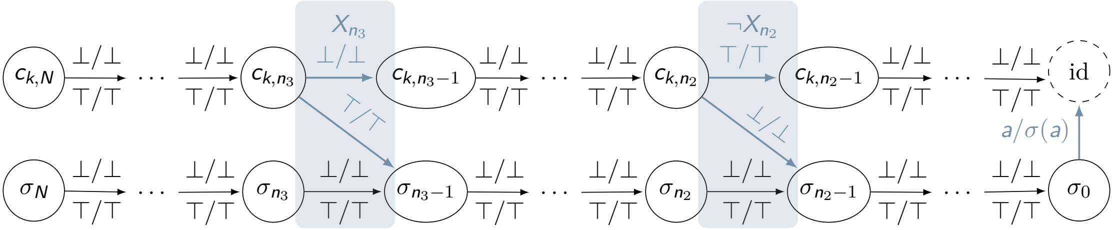
- Important part: Example: $C_k = X_{n_3} \vee \neg X_{n_2} \vee X_{n_1}$ (w. l. o. g.: $n_3 < n_2 < n_1$)



missing transition go to id with b/b

Proof (continued)

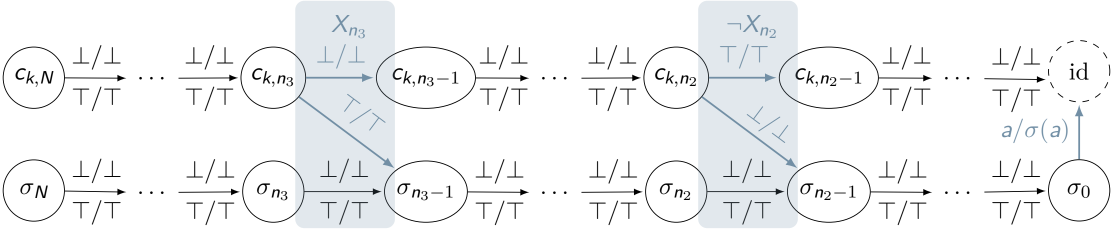
- Important part: **Example:** $C_k = X_{n_3} \vee \neg X_{n_2} \vee X_{n_1}$ (w. l. o. g.: $n_3 < n_2 < n_1$)



missing transition go to id with b/b

Proof (continued)

- Important part: Example: $C_k = X_{n_3} \vee \neg X_{n_2} \vee X_{n_1}$ (w. l. o. g.: $n_3 < n_2 < n_1$)



missing transition go to id with b/b

- Invariant for $w \in \Sigma^N$: $c_k = c_{k,N} \overset{w}{\underset{w}{\longrightarrow}} \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases}$

Proof (continued further)

Invariant for $w \in \Sigma^N$:

$$\begin{tikzcd} & w \arrow[d, -] & \\ c_k \arrow[r, -] & {} \arrow[r] \arrow[d] & \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases} \\ & w & \end{tikzcd}$$

Let $w \in \Sigma^N$.

$$\begin{tikzcd} & w \arrow[d, -] & \\ c_1 \arrow[r, -] & {} \arrow[r] \arrow[d] & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \vdots & \vdots & \vdots \\ & w \arrow[d, -] & \\ c_k \arrow[r, -] & {} \arrow[r] \arrow[d] & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \vdots & \vdots & \vdots \\ & w \arrow[d, -] & \\ c_K \arrow[r, -] & {} \arrow[r] \arrow[d] & \sigma_0 \text{ or } \mathrm{id} \\ & w & \end{tikzcd}$$

Proof (continued further)

Invariant for $w \in \Sigma^N$:

$$c_k \;\begin{array}{c} w \\ \longrightarrow \\ w \end{array}\; \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases}$$

Goal:

φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Let $w \in \Sigma^N$.

$$\begin{array}{cccc}
 & w & & \\
c_1 & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\
 & w & & \\
\vdots & \vdots & \vdots \\
 & w & & \\
c_k & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\
 & w & & \\
\vdots & \vdots & \vdots \\
 & w & & \\
c_K & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\
 & w & &
\end{array}$$

Proof (continued further)

Invariant for $w \in \Sigma^N$:

$$c_k \;\begin{array}{c} w \\ \longrightarrow \\ \downarrow \\ w \end{array}\; \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases}$$

Goal:

φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Set $\boldsymbol{q} = B_N[c_K, \ldots, c_1]$

Convention: B_n uses α_n and β_n instead of α and β

Let $w \in \Sigma^N$.

$$\begin{array}{cccc}
 & w & & \\
c_1 & \longrightarrow & \sigma_0 & \text{or id} \\
 & w & & \\
\vdots & \vdots & & \vdots \\
 & w & & \\
c_k & \longrightarrow & \sigma_0 & \text{or id} \\
 & w & & \\
\vdots & \vdots & & \vdots \\
 & w & & \\
c_K & \longrightarrow & \sigma_0 & \text{or id} \\
 & w & &
\end{array}$$

Proof (continued further)

Invariant for $w \in \Sigma^N$:

$$\begin{array}{ccc} & w & \\ c_k & \longrightarrow & \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases} \\ & w & \end{array}$$

Goal:

φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Set $\boldsymbol{q} = B_N[c_K, \dots, c_1]$

Convention: B_n uses α_n and β_n instead of α and β

Let $w \in \Sigma^N$.

$$\begin{array}{ccc} & w & \\ c_1 & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \vdots & \vdots & \vdots \\ & w & \\ c_k & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \vdots & \vdots & \vdots \\ & w & \\ c_K & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \underbrace{}_{B_N} & & \underbrace{\phantom{\sigma_0 \text{ or } \mathrm{id}}}_{B_0} \end{array}$$

Proof (continued further)

Invariant for $w \in \Sigma^N$:

$$\begin{array}{ccc} & w & \\ c_k & + & \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases} \\ & w & \end{array}$$

Goal:

φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Set $\boldsymbol{q} = B_N[c_K, \ldots, c_1]$

Convention: B_n uses α_n and β_n instead of α and β

Let $w = \langle \mathcal{A} \rangle$ for $\mathcal{A} \models \varphi$.

$$\begin{array}{cccc} & w & & \\ c_1 & + & \sigma_0 \text{ or } \mathrm{id} \\ & w & & \\ \vdots & \vdots & \vdots \\ & w & & \\ c_k & + & \sigma_0 \text{ or } \mathrm{id} \\ & w & & \\ \vdots & \vdots & \vdots \\ & w & & \\ c_K & + & \sigma_0 \text{ or } \mathrm{id} \\ & w & & \end{array}$$

$$\underbrace{[c_1, \ldots, c_K]}_{B_N} \qquad \underbrace{[\sigma_0 \text{ or } \mathrm{id}, \ldots]}_{B_0}$$

Proof (continued further)

Invariant for $w \in \Sigma^N$:

$$\begin{array}{c} w \\ c_k \longrightarrow \\ w \end{array} \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases}$$

Goal:

φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Set $\boldsymbol{q} = B_N[c_K, \dots, c_1] \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Convention: B_n uses α_n and β_n instead of α and β

Let $w = \langle \mathcal{A} \rangle$ for $\mathcal{A} \models \varphi$.

$$\left.\begin{array}{ccc} & w & \\ c_1 & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \vdots & \vdots & \vdots \\ & w & \\ c_k & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \vdots & \vdots & \vdots \\ & w & \\ c_K & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \end{array}\right\} = \begin{array}{c} a_1 \\ \sigma_0 \longrightarrow \mathrm{id} \\ \neq a_1 \end{array}$$

$B_N[$ $\quad B_0$

Proof (continued further)

Invariant for $w \in \Sigma^N$:

$$c_k \overset{w}{\underset{w}{+}} \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases}$$

Goal:

φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Set $\boldsymbol{q} = B_N[c_K, \dots, c_1]$

Convention: B_n uses α_n and β_n instead of α and β

Let $w = \langle \mathcal{A} \rangle$ for $\mathcal{A} \not\models \varphi$.

$$\begin{array}{cccc} & w & & \\ c_1 & + & \sigma_0 \text{ or } \mathrm{id} & \\ & w & & \\ \vdots & \vdots & \vdots & \\ & w & & \\ c_k & + & \sigma_0 \text{ or } \mathrm{id} & \\ & w & & \\ \vdots & \vdots & \vdots & \\ & w & & \\ c_K & + & \sigma_0 \text{ or } \mathrm{id} & \\ B_N & w & B_0 & \end{array}$$

Proof (continued further)

Invariant for $w \in \Sigma^N$:

$$c_k \xrightarrow[w]{w} \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases}$$

Goal:

φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Set $\boldsymbol{q} = B_N[c_K, \ldots, c_1]$

Convention: B_n uses α_n and β_n instead of α and β

Let $w = \langle \mathcal{A} \rangle$ for $\mathcal{A} \not\models \varphi$.

$$\left.\begin{array}{ccc} c_1 & \xrightarrow[w]{w} & \sigma_0 \text{ or } \mathrm{id} \\ \vdots & \vdots & \vdots \\ c_k & \xrightarrow[w]{w} & \sigma_0 \text{ or } \mathrm{id} \\ \vdots & \vdots & \vdots \\ c_K & \xrightarrow[w]{w} & \sigma_0 \text{ or } \mathrm{id} \\ B_N & & B_0 \end{array}\right\} = \mathrm{id} \xrightarrow[u]{u} \mathrm{id}$$

Proof (continued further)

Invariant for $w \in \Sigma^N$:

$$c_k \;\begin{array}{c} w \\ \hline w \end{array}\; \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases}$$

Goal:

φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Set $\boldsymbol{q} = B_N[c_K, \ldots, c_1]$

Convention: B_n uses α_n and β_n instead of α and β

Let $w \notin \{\bot, \top\}^N$.

$$\underbrace{\begin{array}{ccc} & w & \\ c_1 & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \vdots & \vdots & \vdots \\ & w & \\ c_k & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \vdots & \vdots & \vdots \\ & w & \\ c_K & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \end{array}}_{B_N \qquad\qquad B_0}$$

Proof (continued further)

Invariant for $w \in \Sigma^N$:

$$c_k \xrightarrow[w]{w} \begin{cases} \sigma_0 & \text{if } w = \langle \mathcal{A} \rangle, \mathcal{A} \models C_k \\ \mathrm{id} & \text{otherwise} \end{cases}$$

Goal:

φ is satisfiable $\iff$ $\boldsymbol{q} \neq \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$

Set $\boldsymbol{q} = B_N[c_K, \dots, c_1]$

Convention: B_n uses α_n and β_n instead of α and β

Let $w \notin \{\bot, \top\}^N$.

$$\left.\begin{array}{ccc} & w & \\ c_1 & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \vdots & \vdots & \vdots \\ & w & \\ c_k & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ & w & \\ \vdots & \vdots & \vdots \\ & w & \\ c_K & \longrightarrow & \sigma_0 \text{ or } \mathrm{id} \\ B_N & w & B_0 \end{array}\right\} = \mathrm{id} \xrightarrow[u]{u} \mathrm{id}$$

The Uniform Compressed Word Problem for Finitary Automaton Groups

The uniform compressed word problem for finitary automaton groups

Input: a finitary $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
a straight-line program encoding $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$?

The Uniform Compressed Word Problem for Finitary Automaton Groups

The uniform compressed word problem for finitary automaton groups

Input: a finitary $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
a straight-line program encoding $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$?

a context-free grammar generating a single word

The Uniform Compressed Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform compressed word problem for finitary automaton groups

Input: a finitary $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
a straight-line program encoding $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$?

a context-free grammar generating a single word

is $\mathrm{PS{\small PACE}}$-complete.

The Uniform Compressed Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform compressed word problem for finitary automaton groups

Input: a finitary $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
a straight-line program encoding $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$?

is PSPACE-complete.

a context-free grammar generating a single word

…it is ExpSpace-complete for general automaton groups
W., Weiß (2020)

The Uniform Compressed Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform compressed word problem for finitary automaton groups

Input: a finitary $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
a straight-line program encoding $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$?

is PSPACE-complete.

a context-free grammar generating a single word

...it is ExpSpace-complete for general automaton groups
W., Weiß (2020)

- We prove this using a similar reduction form QBF.

The Uniform Compressed Word Problem for Finitary Automaton Groups

Theorem (Kotowsky, W.)

The uniform compressed word problem for finitary automaton groups

Input: a finitary $\mathscr{G}$-automaton $\mathcal{T} = (Q, \Sigma, \delta)$
a straight-line program encoding $\boldsymbol{q} \in (Q^{\pm 1})^*$

Question: is $\boldsymbol{q} = \mathbb{1}$ in $\mathscr{G}(\mathcal{T})$?

is PSPACE-complete.

a context-free grammar generating a single word

…it is ExpSpace-complete for general automaton groups

W., Weiß (2020)

- We prove this using a similar reduction form QBF.
- However: One may also finitely approximate various other groups with PSPACE-complete compressed word problem Bartholdi, Figelius, Lohrey, Weiß (2020)

Thank you!